

Benson type algorithms for linear vector optimization and applications

Andreas H. Hamel · Andreas Löhne · Birgit Rudloff

Received: 20 March 2013 / Accepted: 31 July 2013 / Published online: 24 August 2013
© Springer Science+Business Media New York 2013

Abstract New versions and extensions of Benson’s outer approximation algorithm for solving linear vector optimization problems are presented. Primal and dual variants are provided in which only one scalar linear program has to be solved in each iteration rather than two or three as in previous versions. Extensions are given to problems with arbitrary pointed solid polyhedral ordering cones. Numerical examples are provided, one of them involving a new set-valued risk measure for multivariate positions.

Keywords Vector optimization · Multiple objective optimization · Linear programming · Duality · Algorithms · Outer approximation · Set-valued risk measure · Transaction costs

Mathematics Subject Classification 90C29 · 90C05 · 90-08 · 91G99

1 Introduction

Set-valued approaches to vector optimization are promising in theory and applications. A duality theory in this framework is important for algorithms, and the dual problems can be interpreted in certain applications, see e.g. [10, 13, 15–17, 20, 22, 23, 26, 33]. Benson’s outer approximation algorithm is a fundamental tool for solving linear (and also convex) vector optimization problems [3, 4, 10, 32, 33]. It is also important for solving set-valued problems [28]. Recent applications of linear vector optimization concern financial markets with frictions (transaction costs). For such applications, one obtains optimization problems which are genuinely set-valued. The need to compute the values of a set-valued risk measure for multi-variate random variables was a driving force for this work.

A. H. Hamel
Yeshiva University, New York, USA

A. Löhne (✉)
Martin-Luther-Universität Halle-Wittenberg, Germany
e-mail: andreas.loehne@mathematik.uni-halle.de

B. Rudloff
Princeton University, Princeton, USA

In this article, we introduce a primal and a dual algorithm of Benson type where only one LP has to be solved in each iteration step.¹ In contrast, previous versions [3,4,10,32,33] require at least two different LPs in each step. As the main effort of Benson type algorithms in typical applications is caused by the LPs, the computational time can be reduced considerably by the new algorithms. Another advantage is that all LPs have a very similar structure and therefore the impact of warm starts can be improved. A further benefit is an improvement of the error estimation given in [32,33], i.e., in approximate variants of the algorithms: The same approximation error can be achieved with fewer iteration steps (compare Remark 4.10 and Example 6.1 below). For both the primal and dual algorithm, two variants ('break' and 'no break') are presented and compared (compare Example 6.1). Another novelty of this article is that linear vector optimization problems with arbitrary polyhedral solid pointed ordering cones are treated, whereas in all other references [3,4,9,10,26,32,33] only the special case of the usual ordering cone \mathbb{R}_+^q is considered. This feature will be exploited in applications involving set-valued risk measures for multi-variate random variables in markets with transaction costs. In such situations, ordering cones are usually different from \mathbb{R}_+^q and generated by a large number of directions. A short introduction into this topic and several (numerical) examples are given. Examples 5.2 and 6.6 involve a new type of a set-valued risk measure which we baptized the 'relaxed' worst case risk measure.

This article is organized as follows. In Section 2 we provide some basic notations and results. The next three sections start with short introductions. Section 3 contains an overview on the set-valued approach to linear vector optimization and related duality results where, in contrast to most of the literature, we allow ordering cones more general than \mathbb{R}_+^q . In Section 4 we introduce the new variants of Benson's algorithm. We also give a detailed description of the two-phase-method to treat unbounded problems. Section 5 provides an introduction to applications involving set-valued risk measures, and in Section 6 several numerical examples are reported.

2 Preliminaries

Let $A \subseteq \mathbb{R}^q$. We denote by $\text{cl } A$, $\text{int } A$, $\text{bd } A$ the closure, interior and boundary of A , respectively. The set A is called *solid* if its interior is non-empty. A *convex polyhedron* or a polyhedral convex set A in \mathbb{R}^q is defined to be the intersection of finitely many half spaces, that is

$$A = \bigcap_{i=1}^r \left\{ y \in \mathbb{R}^q \mid (z^i)^T y \geq \gamma_i \right\},$$

where $z^1, \dots, z^r \in \mathbb{R}^q \setminus \{0\}$ and $\gamma_1, \dots, \gamma_r \in \mathbb{R}$. As polyhedra considered in this article are always convex, we will not mention convexity explicitly. Every non-empty polyhedron $A \subseteq \mathbb{R}^q$ can be expressed by means of a finite number of points $y^1, \dots, y^s \in \mathbb{R}^q$ ($s \in \mathbb{N} \setminus \{0\}$) and directions $k^1, \dots, k^t \in \mathbb{R}^q \setminus \{0\}$ ($t \in \mathbb{N}$) through

$$A = \left\{ \sum_{i=1}^s \lambda_i y^i + \sum_{j=1}^t \mu_j k^j \mid \lambda_i \geq 0, i = 1, \dots, s, \sum_{i=1}^s \lambda_i = 1, \mu_j \geq 0, j = 1, \dots, t \right\},$$

where $k \in \mathbb{R}^q \setminus \{0\}$ is called a direction of A if $A + \{\mu \cdot k\} \subseteq A$ for all $\mu > 0$. This can be also written as

$$A = \text{conv} \{y^1, \dots, y^s\} + \text{cone} \{k^1, \dots, k^t\}. \quad (1)$$

¹ A similar variant has been developed independently in [6].

Note that we set $\text{cone } \emptyset = \{0\}$. The polyhedron A is bounded if and only if the cone-part in the above formula is $\{0\}$. The vectors $\{y^1, \dots, y^s\}$ and directions $\{k^1, \dots, k^t\}$ are called the *generators* of A . The set $A_\infty := \text{cone } \{k^1, \dots, k^t\}$ is the recession cone of A . A finite set of half spaces defining a polyhedron A is called *H-representation* (or inequality representation) of A , whereas a finite set of points and directions defining A is called *V-representation* (or generator representation) of A . A bounded polyhedron is called a *polytope*. A convex subset F of a convex set A is called a *face* of A if $(\bar{y}, \hat{y} \in A \wedge \lambda \in (0, 1) \wedge \lambda \bar{y} + (1 - \lambda)\hat{y} \in F)$ implies $y, \hat{y} \in F$. A set F is a *proper* (i.e., $\emptyset \neq F \neq A$) face of a polyhedron A if and only if there is a supporting hyperplane H to A with $F = H \cap A$. The proper $(r - 1)$ -dimensional faces of an r -dimensional polyhedral set A are called *facets* of A . A point $y \in A$ is called a *vertex* of A if $\{y\}$ is a face of A . If $k \in \mathbb{R}^q \setminus \{0\}$ belongs to a half-line face of a polyhedral set A , then k is called *extreme direction* of A .

A polyhedral convex cone $C \subseteq \mathbb{R}^q$ is called *pointed* if it contains no lines. Of course, a solid and pointed convex cone is *non-trivial*, that is, $\{0\} \subsetneq C \subseteq \mathbb{R}^q$. A non-trivial convex pointed cone $C \subseteq \mathbb{R}^q$ defines a partial ordering \leq_C on \mathbb{R}^q by $y^1 \leq_C y^2$ if and only if $y^2 - y^1 \in C$. If $C = \mathbb{R}_+^q := \{y \in \mathbb{R}^q \mid y_1 \geq 0, \dots, y_q \geq 0\}$, then the component-wise ordering $\leq_{\mathbb{R}_+^q}$ is abbreviated to \leq . A point $y \in \mathbb{R}^q$ is called *C-minimal in A* if $y \in A$ and $(\{y\} - C \setminus \{0\}) \cap A = \emptyset$. The set of *C-minimal elements* of A is denoted by $\text{Min}_C A$. If C is additionally solid (but not necessarily pointed), a point $y \in \mathbb{R}^q$ is called *weakly C-minimal in A* if $y \in A$ and $(\{y\} - \text{int } C) \cap A = \emptyset$. Likewise, by replacing C by $-C$, *C-maximal* and *weakly C-maximal* points in a set $A \subseteq \mathbb{R}^q$ are introduced and we write $\text{Max}_C A$ for the set of *C-maximal elements* of A . The dual cone of a cone $C \subseteq \mathbb{R}^q$ is the set $C^+ := \{y^* \in \mathbb{R}^q \mid \forall y \in C : (y^*)^T y \geq 0\}$. The i -th canonical unit vector in \mathbb{R}^q is denoted by e^i .

3 Linear vector optimization

In this section we outline the set-valued approach to linear vector optimization including duality theory and establish a more general setting where arbitrary polyhedral ordering cones $C \subseteq \mathbb{R}^q$ rather than $C = \mathbb{R}_+^q$ are supposed. A comprehensive exposition for the case of the ordering cone $C = \mathbb{R}_+^q$ can be found in [26]. The origin of this approach is discussed in [26, Section 4.8]. A related duality theory and an overview on other approaches to duality for linear vector optimization problems can be found in a recent paper by Luc [29].

3.1 Problem setting and solution concepts

The solution concepts defined in this section are based on the idea that in vector optimization (in contrast to scalar optimization), minimality and infimum attainment are no longer equivalent concepts. In order to have an appropriate complete lattice where an infimum is defined and exists, a set-valued reformulation of the vector optimization problem is necessary. Here we just introduce the solution concepts that result from these ideas. We motivate these concepts from an application oriented viewpoint only. More details and a theoretical motivation can be found in [12, 21, 26].

Let matrices $B \in \mathbb{R}^{m \times n}$, $P \in \mathbb{R}^{q \times n}$, a vector $b \in \mathbb{R}^m$ and a solid pointed polyhedral cone $C \subseteq \mathbb{R}^q$ be given. The following linear vector optimization problem is considered:

$$\text{minimize } P : \mathbb{R}^n \rightarrow \mathbb{R}^q \quad \text{with respect to } \leq_C \quad \text{subject to } Bx \geq b. \quad (\text{P})$$

Define

$$S = \{x \in \mathbb{R}^n \mid Bx \geq b\} \quad \text{and} \quad S^h = \{x \in \mathbb{R}^n \mid Bx \geq 0\}.$$

Of course, we have $S^h = S_\infty$, that is, the non-zero points in S^h are exactly the directions of S . A point $\bar{x} \in S$ is said to be a *minimizer* for (P) if there is no $x \in S$ with $Px \leq_C P\bar{x}$ and $Px \neq P\bar{x}$, that is, $P\bar{x}$ is C -minimal in $P[S] := \{Px \mid x \in S\}$. The set of minimizers of (P) is denoted by $\text{Min}(P)$.

A direction $k \in \mathbb{R}^n \setminus \{0\}$ of S is called a *minimizer* for (P) if the corresponding point $k \in S^h \setminus \{0\}$ is a minimizer of the homogeneous problem

$$\text{minimize } P : \mathbb{R}^n \rightarrow \mathbb{R}^q \text{ with respect to } \leq_C \text{ subject to } Bx \geq 0. \quad (P^h)$$

Definition 3.1 A set $\bar{S} \subseteq S$ is called a set of *feasible* points for (P) and, whenever $S \neq \emptyset$, a set $\bar{S}^h \subseteq S^h \setminus \{0\}$ is called a set of feasible directions for (P).

A pair of sets (\bar{S}, \bar{S}^h) is called a *finite infimizer* for (P) if \bar{S} is a non-empty finite set of feasible points for (P), \bar{S}^h is a (not necessarily non-empty) finite set of feasible directions for (P), and

$$\text{conv } P[\bar{S}] + \text{cone } P[\bar{S}^h] + C = P[S] + C. \quad (2)$$

An infimizer can be understood as a feasible element for a set-valued extension of problem (P) (lattice extension) where the infimum (which is well defined for this lattice extension) is attained, i.e., condition (2) stands for the *infimum attainment*.

The set $\mathcal{P} := P[S] + C$ is called *upper image* of (P). Clearly, if (\bar{S}, \bar{S}^h) is a finite infimizer and $(\{0\}, Y)$ is a V-representation of the cone C , then $(P[\bar{S}], P[\bar{S}^h] \cup Y)$ is a V-representation of the upper image \mathcal{P} .

The following solution concept is based on a combination of minimality and infimum attainment.

Definition 3.2 A finite infimizer (\bar{S}, \bar{S}^h) of (P) is called a *solution* to (P) if \bar{S} and \bar{S}^h consist of only minimizers.

In practice, the upper image \mathcal{P} is one of the most important information for a decision maker. This is due to the fact that in typical applications the dimension n of the decision space is considerably larger than the dimension q of the outcome (or criteria) space. The problem to calculate all the minimizers is usually not tractable. Moreover, the overwhelming set of all minimizers is in general not suitable to support a decision. It is more natural and easier in practice to compare the criteria rather than decisions. Furthermore, also in scalar programming it is often not necessary to know all optimal solution. A solution as introduced above can be seen as an outcome set based concept which provides the information to describe the upper image \mathcal{P} .

3.2 Duality

If vector optimization is considered in a set-valued framework, it is very natural to consider a dual problem with a hyperplane-valued objective function. First, we note that this is also the case in scalar optimization as in \mathbb{R} a point and a hyperplane are the same thing. Secondly, we have in mind the well-known dual description of a convex set by hyperplanes. The upper image \mathcal{P} of a linear vector optimization problem (P) is a convex polyhedron which can be interpreted as an infimum of the lattice extension of problem (P), see [26]. It is therefore natural to ask for a dual description of this convex set which is obtained as the supremum of a suitable dual problem. As a third argument, we want to mention that there is a lack of applications of the classical approaches to duality theory in vector optimization. For instance,

Ehrgott [7] pointed out that “dual algorithms could not be developed because of the absence of a duality theory for MOLP that could be algorithmically exploited”.

The idea of *geometric duality* [20] is to transform the hyperplane-valued dual problem into a vector optimization problem. This idea is taken from the theory of convex polytopes, where an H-representation of a polytope A defines a V-representation of a dual polytope. For instance, if A is solid and contains zero in its interior, an H-representation of the form

$$A = \{x \in \mathbb{R}^q \mid Bx \leq (1, \dots, 1)^T\}$$

exists. The row vectors of the matrix B yield a V-representation of the polar set $A^\circ := \{y^* \in \mathbb{R}^q \mid \forall y \in A : y^{*T}y \leq 1\}$ of A , which is a dual polytope to the polytope A . The duality relation between A and the dual polytope A° is given by an inclusion reversing one-to-one map Ψ between the set of all faces of A and the set of all faces of A° .

A similar duality map can be used to transform a hyperplane-valued optimization problem into a vector-valued problem, which is called the *geometric dual problem*. We assume throughout that there exists a vector

$$c \in \text{int } C \quad \text{such that} \quad c_q = 1, \quad (3)$$

and we fix such a vector c . As C was assumed to be a solid cone, there always exists some $c \in \text{int } C$ such that either $c_q = 1$ or $c_q = -1$. In the latter case we can consider the problem where C and P are replaced by $-C$ and $-P$, which is equivalent to (P) and fulfills (3).

Consider the dual problem

$$\text{maximize } D^* : \mathbb{R}^m \times \mathbb{R}^q \rightarrow \mathbb{R}^q \text{ with respect to } \leq_K \text{ over } T, \quad (D^*)$$

with (linear) objective function

$$D^* : \mathbb{R}^m \times \mathbb{R}^q \rightarrow \mathbb{R}^q, \quad D^*(u, w) := (w_1, \dots, w_{q-1}, b^T u)^T,$$

ordering cone

$$K := \mathbb{R}_+ \cdot (0, 0, \dots, 0, 1)^T,$$

and feasible set

$$T := \{(u, w) \in \mathbb{R}^m \times \mathbb{R}^q \mid u \geq 0, B^T u = P^T w, c^T w = 1, Y^T w \geq 0\},$$

where Y is the matrix whose columns are generators of the ordering cone C .

A point $(\bar{u}, \bar{w}) \in T$ is said to be a *maximizer* for (D^*) if there is no $(u, w) \in T$ with $D^*(u, w) \geq_K D^*(\bar{u}, \bar{w})$ and $D^*(u, w) \neq D^*(\bar{u}, \bar{w})$. The set of maximizers of (D^*) is denoted by $\text{Max}(D^*)$.

Definition 3.3 ([12, 14, 21, 26]). A non-empty finite set \bar{T} of points being feasible for (D^*) is called a *finite supremizer* of (D^*) if

$$\text{conv } D^*[\bar{T}] - K = D^*[T] - K. \quad (4)$$

The set $\mathcal{D}^* = D^*[T] - K$ is called *lower image* of (D^*) . Similarly to above, a finite supremizer yields a V-representation of \mathcal{D}^* . Condition (4) can be interpreted as the attainment of the supremum in a suitable complete lattice, see e.g. [26]. The combination of maximality and supremum attainment leads to a solution.

Definition 3.4 A finite supremizer \bar{T} of (D^*) is called a *solution* to (D^*) if it consists of only maximizers.

Note that, in contrast to (2), we do not need directions in (4). This is due to the simplicity of the cone K in contrast to C .

A duality mapping Ψ for the vector optimization problems (P) and (D*) is now introduced. The bi-affine function

$$\varphi : \mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}, \quad \varphi(y, y^*) := \sum_{i=1}^{q-1} y_i y_i^* + y_q \left(1 - \sum_{i=1}^{q-1} c_i y_i^* \right) - y_q^*$$

is used to define the following two injective hyperplane-valued maps

$$\begin{aligned} H : \mathbb{R}^q &\rightrightarrows \mathbb{R}^q, & H(y^*) &:= \{y \in \mathbb{R}^q \mid \varphi(y, y^*) = 0\}, \\ H^* : \mathbb{R}^q &\rightrightarrows \mathbb{R}^q, & H^*(y) &:= \{y^* \in \mathbb{R}^q \mid \varphi(y, y^*) = 0\}. \end{aligned}$$

The mapping H yields the duality map

$$\Psi : 2^{\mathbb{R}^q} \rightarrow 2^{\mathbb{R}^q}, \quad \Psi(F^*) := \bigcap_{y^* \in F^*} H(y^*) \cap \mathcal{P}.$$

By setting

$$w(y^*) := \left(y_1^*, \dots, y_{q-1}^*, 1 - \sum_{i=1}^{q-1} c_i y_i^* \right) \quad (5)$$

and

$$w^*(y) := (y_1 - y_q c_1, \dots, y_{q-1} - y_q c_{q-1}, -1),$$

we can write

$$\varphi(y, y^*) = w(y^*)^T y - y_q^* = w^*(y)^T y^* + y_q, \quad (6)$$

which is useful for the geometric interpretation of duality.

Weak duality reads as follows.

Theorem 3.5 ([20, 26]). *The following implication is true:*

$$[y \in \mathcal{P} \wedge y^* \in \mathcal{D}^*] \implies \varphi(y, y^*) \geq 0.$$

Note that weak duality implies the inclusions

$$\mathcal{D}^* \subseteq \{y^* \in \mathbb{R}^q \mid \forall y \in \mathcal{P} : \varphi(y, y^*) \geq 0\} \quad \text{and} \quad \mathcal{P} \subseteq \{y \in \mathbb{R}^q \mid \forall y^* \in \mathcal{D}^* : \varphi(y, y^*) \geq 0\},$$

whereas the following strong duality theorem yields even equality.

Theorem 3.6 ([20, 26]). *Let the feasible sets S of (P) and T of (D*) be non-empty. Then*

$$\begin{aligned} [\forall y^* \in \mathcal{D}^* : \varphi(y, y^*) \geq 0] &\implies y \in \mathcal{P}, \\ [\forall y \in \mathcal{P} : \varphi(y, y^*) \geq 0] &\implies y^* \in \mathcal{D}^*. \end{aligned}$$

Remark 3.7 Theorems 3.5 and 3.6 are formulated and proved in the mentioned references only for the special case $C = \mathbb{R}_+^q$ and $c = (1, \dots, 1)^T$. However, using the generalized variants of scalarized problems (P₁(w)), (D₁(w)), (P₂(y)) and (D₂(y)) as stated below, the general case can be obtained in the same way.

The following *geometric duality theorem* provides a third type of duality relation. It takes into account the facial structure of the sets \mathcal{P} and \mathcal{D}^* . Note that geometric duality does not play any role in scalar optimization because the structure of polyhedra in the objective space \mathbb{R} is very simple.

Theorem 3.8 ([19,20]). Ψ is an inclusion reversing one-to-one map between the set of all K -maximal proper faces of \mathcal{D}^* and the set of all proper faces of \mathcal{P} . The inverse map is given by

$$\Psi^{-1}(F) = \bigcap_{y \in F} H^*(y) \cap \mathcal{D}^*.$$

Moreover, if F^* is a K -maximal proper face of \mathcal{D}^* , then

$$\dim F^* + \dim \Psi(F^*) = q - 1.$$

Remark 3.9 The proof of the special case $c = (1, \dots, 1)^T$ and $C = \mathbb{R}_+^q$ can be found in [20]. The general case can be proved in the same way using the generalized versions of $(P_1(w))$, $(D_1(w))$, $(P_2(y))$ and $(D_2(y))$ as defined below. Theorem 3.8 (in the general setting) is also a special case of geometric duality theorem for convex vector optimization problems, see Example 3 in [19].

Remark 3.10 Non- K -maximal proper facets (faces of dimension $q - 1$) of \mathcal{D}^* correspond to the extreme directions of \mathcal{P} by a similar duality relation, where the coupling function φ has to be replaced by $\hat{\varphi} : \mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}$, $\hat{\varphi}(y, y^*) := \varphi(y, y^*) + y_q^*$. The case $C = \mathbb{R}_+^q$, $c = (1, \dots, 1)^T$ has been studied in [26] and the general case is obtained likewise using the generalized variants of $(P_1(w))$, $(D_1(w))$, $(P_2(y))$ and $(D_2(y))$ as defined below.

The following scalarization techniques are fundamental for the algorithms described in the next section. As mentioned in Remarks 3.7, 3.9 and 3.10, they can also be used to prove weak, strong and geometric duality. The *weighted sum* scalarized problem for a parameter vector $w \in C^+$ satisfying $c^T w = 1$ is

$$\min w^T P x \quad \text{subject to } Bx \geq b. \quad (P_1(w))$$

Its dual problem is

$$\max b^T u \quad \text{subject to } \begin{cases} B^T u = P^T w \\ u \geq 0. \end{cases} \quad (D_1(w))$$

The *translative scalarization* (or scalarization by reference variable) is based on problem

$$\min z \quad \text{subject to } \begin{cases} Bx \geq b \\ Z^T P x \leq Z^T y + z \cdot Z^T c, \end{cases} \quad (P_2(y))$$

where Z is the matrix whose columns are the generating directions of the dual cone C^+ of the ordering cone C . The dual program is

$$\max b^T u - y^T Z v \quad \text{subject to } \begin{cases} B^T u - P^T Z v = 0 \\ c^T Z v = 1 \\ (u, v) \geq 0. \end{cases} \quad (\bar{D}_2(y))$$

This problem can be equivalently expressed as

$$\max b^T u - y^T w \quad \text{subject to} \quad \begin{cases} B^T u - P^T w = 0 \\ c^T w = 1 \\ Y^T w \geq 0 \\ u \geq 0, \end{cases} \quad (D_2(y))$$

where Y is the matrix of generating directions of the ordering cone C . The equivalence of $(\bar{D}_2(y))$ and $(D_2(y))$ is a consequence of the following assertion. For vectors $w \in \mathbb{R}^q$, we have

$$Y^T w \geq 0 \iff \forall y \in C : y^T w \geq 0 \iff w \in C^+ \iff \exists v \geq 0 : w = Zv.$$

4 Benson's algorithm and its dual variant

Benson [3,4] motivated his outer approximation algorithm by practical problems that typically involve a huge number of variables and constraints and just a few objective functions. He proposed three advantages of “outcome set based approaches” in comparison to “decision set based approaches”. First, he observed that the set of minimal elements (in the outcome space \mathbb{R}^q) has a simpler structure than the set of minimizers (in the decision space \mathbb{R}^n) because, typically, $q \ll n$. This is beneficial for computational reasons but also for the decision maker. Second, in practice, decision makers prefer to base their decisions (at least in a first stage) on objectives rather than directly on a set of efficient decisions. Third, it is generic that many feasible points are mapped on a single image point which may lead to redundant calculations of “little or no use to the decision maker” [4].

Comparing this motivation with the notions of the previous section, we see that the solution concepts are based on exactly the same motivation (but additionally there is a theoretical motivation, see [21,26]). It is therefore not surprising that the variants of Benson's algorithm presented here just compute solutions in the sense of the previous section.

The dual variant of the algorithm (based on geometric duality) has been established in [8,10]. It was followed by approximating variants [32,33] and by a generalization of the primal algorithm to convex problems [9]. Unbounded problems have been first treated in [26].

Definition 4.1 Problem (P) is said to be *bounded*, if

$$\exists y \in \mathbb{R}^q : P[S] \subseteq \{y\} + C.$$

The generalization from \mathbb{R}_+^q to arbitrary pointed solid polyhedral convex cones is new in this article but has already been used in [27]. We will present simplified variants where only one linear program (rather than two or three) has to be solved in one iteration.²

The idea of the primal algorithm is to evaluate the upper image $\mathcal{P} = P[S] + C$ of problem (P) by constructing appropriate cutting planes. This leads to an iterative refinement of an outer approximation $\mathcal{T} \supseteq \mathcal{P}$ by a decreasing sequence of polyhedral supersets

$$\mathcal{T}^0 \supseteq \mathcal{T}^1 \supseteq \dots \supseteq \mathcal{T}^k = \mathcal{P}.$$

² This simplification was initiated by an idea of Kevin Webster. During a lecture in the Ph.D. course in spring 2011 at ORFE, Princeton University, where the classical variant of Benson's algorithm was introduced, he proposed a variant with the two LPs $(P_2(y))$ and $(D_2(y))$. The advantage over the classical version is that $(P_2(y))$ and $(D_2(y))$ are dual to each other. All further improvements of this article are based on this idea.

Both an H-representation and a V-representation of the approximating supersets \mathcal{T}^i are stored. The algorithm terminates after finitely many steps (say k steps) when the outer approximation coincides with \mathcal{P} .

Unbounded problems are treated by a two-phase method. First, one solves the homogeneous problem (P^h) (which is unbounded, too) and its dual problem

$$\text{maximize } D^{*h} : \mathbb{R}^m \times \mathbb{R}^q \rightarrow \mathbb{R}^q \text{ with respect to } \leq_K \text{ over } T \quad (D^{*h})$$

with objective function

$$D^{*h} : \mathbb{R}^m \times \mathbb{R}^q \rightarrow \mathbb{R}^q, \quad D^{*h}(u, w) := (w_1, \dots, w_{q-1}, 0)^T.$$

To this end, (P^h) is transformed into an equivalent bounded problem

$$\text{minimize } P : \mathbb{R}^n \rightarrow \mathbb{R}^q \text{ with respect to } \leq_C \text{ subject to } Bx \geq 0, \quad \eta^T Px \leq 1, \quad (P^\eta)$$

where $\eta \in \text{int}(\mathcal{D}^{*h} + K)$ with $c^T \eta = 1$ (\mathcal{D}^{*h} denotes the lower image of (D^{*h})).

In the second phase, a primal and dual solution of the homogeneous problem (P_h) are used to calculate a primal and dual solution of the original (inhomogeneous and unbounded) problem (P) . The two-phase method requires an algorithm that works whenever an H-representation of an initial outer approximation $\mathcal{T}^0 \supseteq \mathcal{P}$ with $\mathcal{T}_\infty^0 = \mathcal{P}_\infty$ is known. If an H-representation of \mathcal{P}_∞ is known, that is

$$\mathcal{P}_\infty = \left\{ y \in \mathbb{R}^q \mid (w^i)^T y \geq 0, \quad i = 1, \dots, r \right\},$$

and if γ_i denotes the optimal value of $(P_1(w^i))$ for $i = 1, \dots, r$, then

$$\mathcal{T}^0 = \left\{ y \in \mathbb{R}^q \mid (w^i)^T y \geq \gamma_i, \quad i = 1, \dots, r \right\}$$

is the desired outer approximation of \mathcal{P} satisfying $\mathcal{T}_\infty^0 = \mathcal{P}_\infty$.

If problem (P) is bounded, we have $C = \mathcal{P}_\infty$, i.e., an H-representation of the ordering C is required. Otherwise, whenever (P) is feasible, the upper image $\mathcal{P}^h := P[S^h] + C$ of the homogeneous problem (P^h) coincides with \mathcal{P}_∞ . By geometric duality, a dual solution to (P^h) yields an H-representation of $\mathcal{P}^h = \mathcal{P}_\infty$.

The idea of such an algorithm can be explained geometrically. Let $\mathcal{T}^0 \supseteq \mathcal{P}$ be an initial outer representation of \mathcal{P} , i.e., $\mathcal{T}_\infty^0 = \mathcal{P}_\infty$. First, the vertices of \mathcal{T}^0 are computed from its H-representation. This can be realized by *vertex enumeration*, which is a standard method in Computational Geometry, see e.g. [2, 5]. Secondly, for a vertex $t^0 \in \mathcal{T}^0$, problem $(P_2(t^0))$ is solved. Usually, LP solvers yield simultaneously a solution of both the primal and the dual problem. If the optimal value of $(P_2(t^0))$ is zero, then t^0 belongs to \mathcal{P} and one proceeds with the next vertex of \mathcal{T}^0 . If every vertex of \mathcal{T}^0 belongs to \mathcal{P} , we have $\mathcal{T}^0 = \mathcal{P}$. Otherwise, for $t^0 \notin \mathcal{P}$, a solution of $(P_2(t^0))$ yields a point $s^0 \in \text{bd } \mathcal{P} \cap \text{int } \mathcal{T}^0$, see Proposition 4.2 below. The solution of the dual problem, defines a supporting hyperplane H^0 of \mathcal{P} that contains s^0 . The corresponding halfspace H_+^0 contains \mathcal{P} but not t^0 . An H-representation of an improved outer approximation $\mathcal{T}^1 := \mathcal{T}^0 \cap H_+^0$ is obtained immediately. This procedure is repeated until, after finitely many steps, $\mathcal{T}^k = \mathcal{P}$. A solution (\bar{S}, \bar{S}^h) of (P) is obtained by collecting those points x that arise during the procedure from a solution (x, z) of $(P_2(t^i))$ with zero optimal value. In this case we have $t = Px$ for some vertex t of \mathcal{T}^i . Hence Px is a vertex of \mathcal{P} which implies that x is a minimizer for (P) . In the unbounded case, \bar{S}^h contains directions that originate from a solution to the homogeneous problem. A solution of the dual vector optimization problem (D^*) is obtained by collecting those dual solutions of $(P_2(t^i))$ with non-zero optimal value.

Proposition 4.2 Let $S \neq \emptyset$, $C \subseteq \mathbb{R}^q$ a solid pointed polyhedral cone and let $c \in \text{int } C$. For every $t \in \mathbb{R}^q$, there exist optimal solutions (\bar{x}, \bar{z}) to $(P_2(t))$ and (\bar{u}, \bar{w}) to $(D_2(t))$. Each solution (\bar{u}, \bar{w}) to $(D_2(t))$ defines a supporting hyperplane $H := \{y \in \mathbb{R}^q \mid \bar{w}^T y = b^T \bar{u}\}$ of \mathcal{P} such that $s := t + \bar{z} \cdot c \in H \cap \mathcal{P}$. We have

$$t \notin \mathcal{P} \iff \bar{z} > 0, \quad t \in \text{wMin } \mathcal{P} \iff \bar{z} = 0, \quad t \in \text{int } \mathcal{P} \iff \bar{z} < 0.$$

Proof Fix $t \in \mathbb{R}^q$. Since $S \neq \emptyset$ and $c \in \text{int } C$, $(P_2(t))$ is feasible. Assuming $(P_2(t))$ is not bounded, we obtain $t + (z - n)c - Px \in C$ for all $n \in \mathbb{N}$. Dividing by n and letting $n \rightarrow \infty$, we conclude $-c \in C$. As $c \in \text{int } C$, convexity of C implies $0 \in \text{int } C$. Thus $C = \mathbb{R}^q$, a contradiction. Consequently, $(P_2(t))$ and, by duality, also $(D_2(t))$ have optimal solutions (\bar{x}, \bar{z}) and (\bar{u}, \bar{w}) , respectively. Duality yields $b^T \bar{u} - t^T \bar{w} = \bar{z}$ and thus $s = t + \bar{z}c$ belongs to H . Of course, H is a hyperplane as the constraint $\bar{w}^T c = 1$ of $(D_2(t))$ implies $\bar{w} \neq 0$. From $P\bar{x} \leq_C t + \bar{z} \cdot c$ we conclude that $s = t + \bar{z}c$ belongs to \mathcal{P} . For arbitrary $y \in \mathcal{P}$, there exists $x \in S$ such that $y \geq_C Px$. Hence $(x, 0)$ is feasible for $(P_2(y))$. Weak duality between $(P_2(y))$ and $(D_2(y))$ implies that $y^T w \geq b^T u$ for every $(u, w) \in T$, in particular, $y^T \bar{w} \geq b^T \bar{u}$. Hence $H = \{y \in \mathbb{R}^q \mid y^T \bar{w} = b^T \bar{u}\}$ is a supporting hyperplane to \mathcal{P} . The remaining statements are now obvious, where the fact $\text{wMin } \mathcal{P} = \text{bd } \mathcal{P}$ can be used. \square

Proposition 4.3 Every vertex of \mathcal{P} is minimal.

Proof Let $y \in \mathcal{P} = P[S] + C$ be not minimal for \mathcal{P} . Then there are $z \in \mathcal{P}$ and $k \in C \setminus \{0\}$ such that $z = y - k$. The points $y - k$ and $y + k$ belong to \mathcal{P} and we have $y = \frac{1}{2}(y - k) + \frac{1}{2}(y + k)$. Hence y is not a vertex of \mathcal{P} . \square

Two functions are used in the following algorithm. The function $\text{dual}()$ computes a V-representation of an outer approximation \mathcal{T} from an H-representation of \mathcal{T} , i.e., this function consists essentially of vertex enumeration. This H-representation of \mathcal{T} , however, is stored in a dual format, namely, as a V-representation of an inner approximation \mathcal{T}^* of the lower image \mathcal{D}^* of (\mathcal{D}^*) , where

$$\mathcal{T}^* = \{y^* \in \mathbb{R}^q \mid \varphi(y, y^*) \geq 0, y \in \mathcal{T}\}. \quad (7)$$

The following duality relation holds.

Proposition 4.4 If $\emptyset \neq \mathcal{T} \subsetneq \mathbb{R}^q$ is closed and convex and $\mathcal{T}_\infty \supseteq C$, then

$$\mathcal{T} = \{y \in \mathbb{R}^q \mid \varphi(y, y^*) \geq 0, y^* \in \mathcal{T}^*\}. \quad (8)$$

Proof The inclusion \subseteq is obvious. Assume that the inclusion \supseteq does not hold, i.e., there exists $\bar{y} \in \mathbb{R}^q \setminus \mathcal{T}$ with $\varphi(\bar{y}, y^*) \geq 0$ for all $y^* \in \mathcal{T}^*$. Applying separation arguments, we get $\eta \in C^+ \setminus \{0\}$ with $\eta^T \bar{y} < \inf_{y \in \mathcal{T}} \eta^T y =: \gamma$. By (3), we can assume $\eta^T c = 1$. Setting $\bar{y}^* := (\eta_1, \dots, \eta_{q-1}, \gamma)$, we get $w(\bar{y}^*) = \eta$ and $\varphi(y, \bar{y}^*) = w(\bar{y}^*) - \bar{y}_q^* = \eta^T y - \gamma$. For all $y \in \mathcal{T}$, we have $\eta^T y - \gamma \geq 0$, i.e., $\bar{y}^* \in \mathcal{T}^*$. But $\varphi(\bar{y}, \bar{y}^*) = \eta^T \bar{y} - \gamma < 0$, which is a contradiction. \square

In the following algorithm, a V-representation of a polyhedron \mathcal{T} (that contains no lines) is denoted by $(\mathcal{T}_{\text{points}}, \mathcal{T}_{\text{dirs}})$, i.e., $\mathcal{T} = \text{conv } \mathcal{T}_{\text{points}} + \text{cone } \mathcal{T}_{\text{dirs}}$. We assume further that a V-representation returned by the function $\text{dual}()$ is minimal, i.e., $\mathcal{T}_{\text{points}}$ consists of only vertices of \mathcal{T} and $\mathcal{T}_{\text{dirs}}$ consists of only extreme directions of \mathcal{T} .

The function $\text{solve}()$ returns an optimal solution (x, z) of $(P_2(t))$ and an optimal solution (u, w) of $(D_2(t))$. Since $(D_2(t))$ is, up to a substitution, the dual program of $(P_2(t))$, only one linear program has to be solved.

The variables in the following algorithm are arrays of vectors. By $|A|$ we denote the number of vectors in an array A and by $A[i]$ we refer to the i th vector in A . The command *break* exits the inner-most loop.

Algorithm 1.

Input:

B, b, P, Z (data of (P));
a solution $(\{0\}, \bar{S}^h)$ to (P^h) ;
a solution \bar{T}^h to (D^{*h}) ;

Output:

(\bar{S}, \bar{S}^h) ... a solution to (P) ;
 \bar{T} ... a solution to (D^*) ;
 $(\mathcal{T}_{points}, \mathcal{T}_{dirs})$... a V-representation of \mathcal{P} ;
 $(\mathcal{T}_{points}^*, \{-e^q\})$... a V-representation of \mathcal{D}^* ;

$\bar{T} \leftarrow \{(\text{solve}(D_1(w)), w) \mid (u, w) \in \bar{T}^h\}$;

repeat

$flag \leftarrow 0$;

$\bar{S} \leftarrow \emptyset$;

$\mathcal{T}_{points}^* \leftarrow \{D^*(u, w) \mid (u, w) \in \bar{T}\}$;

$(\mathcal{T}_{points}, \mathcal{T}_{dirs}) \leftarrow \text{dual}(\mathcal{T}_{points}^*, \{-e^q\})$;

for $i = 1$ **to** $|\mathcal{T}_{points}|$ **do**

$t \leftarrow \mathcal{T}_{points}[i]$;

$(x, z, u, w) \leftarrow \text{solve}(P_2(t)/D_2(t))$;

if $z > 0$ **then**

$\bar{T} \leftarrow \bar{T} \cup \{(u, w)\}$;

$flag \leftarrow 1$;

break; (optional)

else

$\bar{S} \leftarrow \bar{S} \cup \{x\}$;

end if;

end for;

until $flag = 0$;

Theorem 4.5 Let $S \neq \emptyset$, suppose \mathcal{P}^h has a vertex and assume that the command

$$(\mathcal{T}_{points}, \mathcal{T}_{dirs}) \leftarrow \text{dual}(\mathcal{T}_{points}^*, \mathcal{T}_{dirs}^*)$$

generates a minimal V-representation of \mathcal{T} from a given V-representation of \mathcal{T}^* according to (7). Then Algorithm 1 is correct and finite.

Proof As \bar{T}^h is non-empty (by the definition of a solution), we can choose some $(u, w) \in \bar{T}^h$. Then $D^{*h}(u, w) = (w_1, \dots, w_{q-1}, 0)$ is K -maximal in \mathcal{D}^{*h} . Hence u solves the homogeneous variant (i.e., we set $b = 0$) of $(D_1(w))$. Consequently, $(D_1(w))$ (for arbitrary b) is feasible. Since $S \neq \emptyset$, $(P_1(w))$ is feasible, too. Thus, by linear programming duality, $(D_1(w))$ has a solution.

The set $\mathcal{T}^* := \text{conv } \mathcal{T}_{points}^* + \text{cone } \{-e^q\}$ is a non-empty subset of \mathcal{D}^* . Hence, by Theorem 3.5, after calling the function $\text{dual}()$, $\mathcal{T} := \text{conv } \mathcal{T}_{points} + \text{cone } \mathcal{T}_{dirs}$ is a superset of \mathcal{P} .

As \bar{T}^h solves the dual of the homogeneous problem, we have $\mathcal{T}_\infty = \mathcal{P}_\infty = \mathcal{P}^h$, see [26, Section 4.6] for more details. As \mathcal{P}^h is assumed to have a vertex, \mathcal{T} must have a vertex, hence the array \mathcal{T}_{points} is non-empty.

By Proposition 4.2, solutions to $(P_2(t))$ and $(D_2(t))$ exist. The vectors $x \in \bar{S}$ are minimizers of (P). Indeed, x is added to \bar{S} only if $z = 0$. In this case, we have $t \in \mathcal{P}$, where t is a vertex of $\mathcal{T} \supseteq \mathcal{P}$ because, by assumption, \mathcal{T}_{points} contains only vertices of \mathcal{T} . Hence t is a vertex of \mathcal{P} and, by Proposition 4.3, t is a minimizer for (P).

The algorithm terminates if all vertices of \mathcal{T} belong to \mathcal{P} . Since $\mathcal{P}_\infty = \mathcal{T}_\infty$, we conclude $\mathcal{P} = \mathcal{T}$, i.e., (\bar{S}, \bar{S}^h) is an infimizer of (P) and $(\mathcal{T}_{points}, \mathcal{T}_{dirs})$ is a V-representation of \mathcal{P} .

A solution (u, w) to $(D_2(t))$ is always a maximizer of (D^*) , i.e., \bar{T} consists of only maximizers. Since at termination $\mathcal{T} = \mathcal{P}$, Theorem 3.6 implies $\mathcal{T}^* = \mathcal{D}^*$ and thus \bar{T} is a supremizer for (D^*) and $(\mathcal{T}_{points}^*, \{-e_q\})$ is a V-representation of \mathcal{D}^* .

Finally we show that the algorithm terminates after a finite number of steps. The point $s^k := t^k + z^k \cdot c$ computed in iteration k (consider the ‘repeat’ loop) by solving $(P_2(t^k)/D_2(t^k))$ belongs to $\text{int } \mathcal{T}^{k-1}$ whenever $z^k > 0$. We have $\mathcal{T}^k := \mathcal{T}^{k-1} \cap \{y \in \mathbb{R}^q \mid (w^k)^T y \geq (u^k)^T b\}$ and by Proposition 4.2 we know that $F := \{y \in \mathcal{P} \mid (w^k)^T y = (u^k)^T b\}$ is a face of \mathcal{P} with $s^k \in F$, where $F \subseteq \text{bd } \mathcal{T}^k$. This means for the next iteration that $s^{k+1} \notin F$ (because $s^{k+1} \in \text{int } \mathcal{T}^k$), and therefore s^{k+1} belongs to another face of \mathcal{P} . Since \mathcal{P} is polyhedral, it has a finite number of faces, hence the algorithm is finite. \square

We now turn to the dual variant of Algorithm 1. An analogous construction is now applied to the lower image \mathcal{D}^* , i.e., a finite sequence of polyhedral sets

$$\mathcal{T}^{*0} \supseteq \mathcal{T}^{*1} \supseteq \dots \supseteq \mathcal{T}^{*k} = \mathcal{D}^*$$

is calculated. Using the upper image \mathcal{P}^h (which is a polyhedral cone) of the homogeneous problem (P^h) , we define the set

$$\Delta := \left\{ y^* \in \mathbb{R}^q \mid w(y^*) \in (\mathcal{P}^h)^+ \right\}.$$

The counterpart of Proposition 4.2 is the following.

Proposition 4.6 *Let $S \neq \emptyset$ and $t^* \in \Delta$. For $w := w(t^*)$, $(P_1(w))$ has a solution and for every such solution \bar{x} , $H^*(P\bar{x})$ is a supporting hyperplane of \mathcal{D}^* that contains*

$$s^* := (t_1^*, \dots, t_{q-1}^*, w^T P\bar{x}) \in \text{Max}_K \mathcal{D}^*. \quad (9)$$

Moreover, we have

$$t^* \notin \mathcal{D}^* \iff w^T P\bar{x} < t_q^*, \quad t^* \in \text{Max}_K \mathcal{D}^* \iff w^T P\bar{x} = t_q^*.$$

Proof Since $t^* \in \Delta$, for all $k \in \mathcal{P}^h$, we have $w^T k \geq 0$. This means that the homogeneous variant of the linear program $(P_1(w))$ (i.e., we set $b = 0$ in $(P_1(w))$) is bounded (and feasible, as 0 is feasible). Consequently, the dual program is feasible, even for arbitrary b , i.e., $(D_1(w))$ is feasible. On the other hand, $(P_1(w))$ is feasible, since we assumed $S \neq \emptyset$. Altogether this implies that both $(P_1(w))$ and $(D_1(w))$ have optimal solutions denoted, respectively, by \bar{x} and \bar{u} . Strong duality implies $w^T P\bar{x} = b^T \bar{u}$. Thus, (9) holds. We have $s^* \in H^*(P\bar{x})$ because this can be written as $w(s^*)^T P\bar{x} = s_q^*$ where we have $w = w(t^*) = w(s^*)$. Together with Theorem 3.5, we obtain that $H^*(P\bar{x})$ is a supporting hyperplane of \mathcal{D}^* that contains s^* . The remaining statements are now obvious. \square

The following consequence of Proposition 4.4 is useful to characterize the condition $t^* \in \Delta$.

Corollary 4.7 *Let the assumptions of Proposition 4.4 be satisfied. Then, $w(y^*) \in (\mathcal{T}_\infty)^+$ for all $y^* \in \mathcal{T}^*$.*

Proof Assuming the contrary, there is $\bar{y}^* \in \mathcal{T}^*$ and $k \in \mathcal{T}_\infty$ with $w(y^*)^T k < 0$. Let $\bar{y} \in \mathcal{T}$. For sufficiently large $\lambda > 0$, using (6), we obtain $\varphi(\bar{y} + \lambda k, \bar{y}^*) < 0$, which contradicts Proposition 4.4. \square

The following dual algorithm has the same input and output as Algorithm 1. Similar functions are used. The function *dual()* computes a V-representation of an outer approximation \mathcal{T}^* of \mathcal{D}^* from a V-representation of an inner approximation \mathcal{T} of \mathcal{P} . In contrast to Algorithm 1, it is not necessary that *dual()* returns a minimal V-representation. The recession cone of sets \mathcal{T}^* occurring in the algorithm is known, in fact, we always have $\mathcal{T}_\infty^* = -K = \mathbb{R}_+ \{-e^q\}$. Therefore we denote the return of the function *dual()* by $(\mathcal{T}_{points}^*, \sim)$ indicating that the second return value (the array containing the extreme directions of \mathcal{T}^*) is not used.

The function *solve()* returns an optimal solution x of $(P_1(w))$ and an optimal solution u of $(D_1(w))$. Again, only one linear program has to be solved.

Algorithm 2.

Input:

B, b, P, Y (data of Problem (P));
a solution $(\{0\}, \bar{S}^h)$ to (P^h) ;
a solution \bar{T}^h to (D^{*h}) ;

Output:

(\bar{S}, \bar{S}^h) is a solution to (P);
 \bar{T} is a solution to (D^*) ;
 $(\mathcal{T}_{points}, \mathcal{T}_{dirs}) \dots$ a V-representation of \mathcal{P} ;
 $(\mathcal{T}_{points}^*, \{-e^q\}) \dots$ a V-representation of \mathcal{D}^* ;

$\mathcal{T}_{dirs} \leftarrow \{Px \mid x \in \bar{S}^h\} \cup \{y \mid y \text{ is a column of } Y\}$;

$\bar{w} \leftarrow \text{mean}\{w \mid (u, w) \in \bar{T}^h\}$;

$\bar{S} \leftarrow \{\text{solve}(P_1(\bar{w}))\}$;

repeat

$flag \leftarrow 0$;

$\bar{T} \leftarrow \emptyset$;

$\mathcal{T}_{points} \leftarrow \{Px \mid x \in \bar{S}\}$

$(\mathcal{T}_{points}^*, \sim) \leftarrow \text{dual}(\mathcal{T}_{points}, \mathcal{T}_{dirs})$;

for $i = 1$ **to** $|\mathcal{T}_{points}^*|$ **do**

$t^* \leftarrow \mathcal{T}_{points}^*[i]$;

$w \leftarrow w(t^*)$;

$(x, u) \leftarrow \text{solve}(P_1(w)/D_1(w))$;

if $t_q^* - b^T u > 0$ **then**

$\bar{S} \leftarrow \bar{S} \cup \{x\}$;

$flag \leftarrow 1$;

break; (optional)

else

$\bar{T} \leftarrow \bar{T} \cup \{(u, w)\}$;

end if;

end for;

until $flag = 0$;
 delete points $x \in \tilde{S}$ whenever Px is not a vertex of \mathcal{T} ;

Remark 4.8 The last line in the algorithm is easy to realize, for instance, by computing a minimal V-representation using the command

$$(\mathcal{T}_{points}, \mathcal{T}_{dirs}) \leftarrow \text{dual}(\mathcal{T}_{points}^*, \mathcal{T}_{dirs}^*)$$

from Algorithm 1 by standard vertex enumeration methods. Then one has to test if for $x \in \tilde{S}$, Px belongs to \mathcal{T}_{points} , if not, x is deleted from \tilde{S} . In particular, it is not necessary to solve a linear program.

Theorem 4.9 Let $S \neq \emptyset$ and assume that \mathcal{P}^h has a vertex. Then, Algorithm 2 is correct and finite.

Proof By similar arguments as in the proof of Theorem 4.5 one can show that $P_1(\bar{w})$ has a solution.

The set $\mathcal{T} := \text{conv } \mathcal{T}_{points} + \text{cone } \mathcal{T}_{dirs}$ is a subset of \mathcal{P} . Hence, by Theorem 3.5, after calling the function $\text{dual}()$, $\mathcal{T}^* := \text{conv } \mathcal{T}_{points}^* + \text{cone } \mathcal{T}_{dirs}^*$ is a superset of \mathcal{D}^* . Since $\mathcal{T} \neq \mathbb{R}^q$, $c \in \text{int } C$ and $\mathcal{T}_\infty \supseteq C$, we have $\text{cone } \mathcal{T}_\infty^* = \mathbb{R}_+ \cdot \{-e^q\}$, i.e., we can set $\mathcal{T}_{dirs}^* = \{-e^q\}$ and we know that $\mathcal{T}_{points}^* \neq \emptyset$.

The array $(\{0\}, \mathcal{T}_{dirs})$ provides a V-representation of \mathcal{P}^h , i.e., $\mathcal{T}_\infty = \mathcal{P}^h$. Corollary 4.7 yields that $\mathcal{T}_{points}^* \subseteq \Delta$. Hence, by Proposition 4.6, solutions to $(P_1(w))$ and $(D_1(w))$ exist. It can be easily shown that the vectors $(u, w) \in \bar{T}$ are maximizers of (D^*) , see also [26, Lemma 4.51].

The algorithm terminates, if $\mathcal{T}_{points}^* \subseteq \mathcal{D}^*$. Since $\mathcal{D}_\infty^* = \mathcal{T}_\infty^* = \mathbb{R}_+ \{-e^q\}$, we conclude $\mathcal{D}^* = \mathcal{T}^*$, i.e., \bar{T} is a supremizer of (D^*) and $(\mathcal{T}_{points}^*, \{-e^q\})$ is a V-representation of \mathcal{D}^* . Since at termination $\mathcal{T}^* = \mathcal{D}^*$, Theorem 3.6 implies $\mathcal{T} = \mathcal{P}$. Thus $(\mathcal{T}_{points}, \mathcal{T}_{dir})$ is a (not necessarily minimal) V-representation of \mathcal{P} . A solution x to $(P_1(w))$ is in general not a minimizer for (P) (but only “weakly efficient”, compare e.g. [26, Theorem 4.1]). Therefore, in the last line of the algorithm, x is deleted from \tilde{S} , whenever Px is not a vertex of \mathcal{T} . According to Proposition 4.3, the remaining set \tilde{S} consists of only minimizers. It is non-empty because, by assumption, \mathcal{P}^h has a vertex and hence $\mathcal{T} = \mathcal{P}$ must have a vertex. As non-vertex points are redundant in a V-representation of a set which has a vertex, the property of \tilde{S} being an infimizer for (P) is maintained by deleting the non-minimizers in \tilde{S} .

Finally we show that the algorithm terminates after finitely many steps. We consider the ‘repeat’ loop in iteration k . We set $w = w^k$ and $t^* = t^{*k}$ and denote the solutions of $(P_1(w^k))$ and $(D_1(w^k))$ by x^k and u^k , respectively. The point $s^{*k} := t^{*k} + z^{*k} \cdot \{-e^q\}$, where $z^{*k} := (t_q^{*k} - b^T u^k)$ belongs to $\mathcal{T}^{*(k-1)} \setminus \text{Max}_K \mathcal{T}^{*(k-1)}$ whenever $z^{*k} > 0$. We have $\mathcal{T}^{*k} := \mathcal{T}^{*(k-1)} \cap \{y^* \in \mathbb{R}^q \mid \varphi(Px^k, y^*) \geq 0\}$ and by Proposition 4.6 we know that $F^* := \{y^* \in \mathcal{D}^* \mid \varphi(Px^k, y^*) = 0\}$ is a face of \mathcal{D}^* with $s^{*k} \in F^*$. Likewise to [26, Lemma 4.48], we see that $F^* \subseteq \text{Max}_K \mathcal{T}^{*k}$. This means for the next iteration that $s^{*(k+1)} \notin F^*$ (because $s^{*(k+1)} \in \mathcal{T}^{*k} \setminus \text{Max}_K \mathcal{T}^{*k}$), and therefore $s^{*(k+1)}$ belongs to another face of \mathcal{D}^* . Since \mathcal{D}^* is polyhedral, it has a finite number of faces, hence the algorithm is finite. \square

Let us summarize the two-phase method for solving unbounded problems. We consider an arbitrary linear vector optimization problem, where we only assume that C is a solid pointed polyhedral cone. We fix some c according to (3), which is always possible in the way described after (3). In phase 1, we first try to compute some $\eta \in \text{int}(\mathcal{D}^* + K)$ with $\eta^T c = 1$. This can be realized by Algorithm 3 in [26, Section 5.5], where the set T has to be adapted

to the more general setting of this article. Note that c has a different meaning in [26, Section 5.5]. The first LP solved by the mentioned algorithm is

$$\min 0^T w + 0^T u \quad \text{s.t.} \quad (u, w) \in T.$$

If this linear program is infeasible, then (D^*) is infeasible. Otherwise one obtains either some $\eta \in \text{int}(\mathcal{D}^* + K)$ or the algorithm indicates that $\text{int}(\mathcal{D}^* + K)$ is empty. In the latter case, we know that \mathcal{P}^h has no vertex. This means that \mathcal{P} , if non-empty, contains a line. This case has not been treated so far. Since $c_q = 1$ according to (3), the condition $\eta^T c = 1$ can be always realized by an appropriate choice of η_q .

Next, we solve (P^η) by either Algorithms 1 or 2. Since (P^η) is bounded, a solution of the primal and dual homogeneous problem of (P^η) can be easily obtained. However, this is not necessary as the u -components of $(u, w) \in \bar{T}^h$ are not used in Algorithms 1 and 2. Therefore we can use

$$\bar{S}^h = \emptyset \quad \text{and} \quad \bar{T}^h = \left\{ \left(0, \frac{z}{z^T c} \right) \mid z \text{ is a column of } Z \right\}$$

as an input of Algorithms 1 or 2 to solve (P^η) , compare also [26, Theorem 5.20]. Let $(\bar{S}_\eta, \bar{S}_\eta^h)$ be a solution of (P^η) and let \bar{T}_η be a solution of the dual problem of (P^η) . Then, a solution of (P_h) is obtained by setting

$$(\bar{S}, \bar{S}^h) := (\{0\}, \bar{S}_\eta \setminus \{0\}),$$

compare [26, Theorem 5.23]. A solution \bar{T}^h of (D^{*h}) can be obtained from \bar{T}_η but again only the w -components are required by Algorithms 1 or 2 in phase 2. As a consequence of [26, Theorem 5.25], we can use

$$\bar{T}^h := \left\{ (0, w(y^*)) \mid y^* \in \mathcal{T}_{points}^*, y_q^* = 0 \right\}$$

as an input of Algorithms 1 or 2 in the second phase, where \mathcal{T}_{points}^* is the result from the first phase, i.e., $(\mathcal{T}_{points}^*, \{-e^q\})$ is a V-representation of the lower image $\mathcal{D}^{*\eta}$ of the dual problem of (P^η) . In the second phase, the first LP to be solved in Algorithm 1 is $(D_1(w))$. If $(D_1(w))$ turns out to be unbounded, we know that (P) is infeasible. Likewise, if the first LP in Algorithm 2, namely $(P_1(\bar{w}))$, is infeasible, we know that (P) is infeasible. Otherwise, according to Theorems 4.5 and 4.9, solutions of (P) and (D^*) are computed.

Remark 4.10 In practice the condition $z > 0$ in Algorithm 1 is replaced by $z > \varepsilon$ for some $\varepsilon > 0$. Assume that the results of the first phase are always exact. Then, in the second phase, Algorithm 1 yields an ε -solution of (P) in the sense that in Definition 3.2 the finite infimizer is replaced by a finite ε -infimizer, i.e., condition (2) is replaced by

$$\text{conv } P[\bar{S}] + \text{cone } P[\bar{S}^h] + C - \varepsilon\{c\} \supseteq P[S] + C. \quad (10)$$

Taking into account that (using the assumption $c_q = 1$ in (3))

$$\varphi(y - \varepsilon c, y^*) = \varphi(y, y^* + \varepsilon e^q),$$

we see that Algorithm 1 yields an ε -solution of (D^*) in the sense that in Definition 3.4 a finite supremizer is replaced by a finite ε -supremizer, i.e., condition (4) is replaced by

$$\text{conv } D^*[\bar{T}] - K + \varepsilon\{e^q\} \supseteq D^*[T] - K. \quad (11)$$

Likewise, in Algorithm 2 the condition $t_q^* - b^T u > 0$ is replaced by $t_q^* - b^T u > \varepsilon$ for some $\varepsilon > 0$. Consequently, Algorithm 2 yields an ε -solution of (D^*) . It also yields an ε -infimizer

of (P), but in general not an ε -solution of (P). The reason is that the last line in Algorithm 2 only works for the exact algorithm.

Note further that an ε -solution (\bar{S}, \bar{S}^h) of (P) refers to an inner and an outer approximation of the upper image \mathcal{P} in the sense that

$$\text{conv } P[\bar{S}] + \text{cone } P[\bar{S}^h] + C \subseteq \mathcal{P} \subseteq \text{conv } P[\bar{S}] + \text{cone } P[\bar{S}^h] + C - \varepsilon\{c\}.$$

Likewise, an ε -solution \bar{T} of (D^*) refers to an inner and an outer approximation of the lower image \mathcal{D}^* in the sense that

$$\text{conv } D^*[\bar{T}] - K \subseteq \mathcal{D}^* \subseteq \text{conv } D^*[\bar{T}] - K + \varepsilon\{e^q\}.$$

Note that the approximation error of the classical variant of Benson's algorithm and its dual variant has been studied in [32, 33].

5 Computation of polyhedral set-valued risk measures

Set-valued risk measures evaluate the risk of multi-variate random portfolios $X: \Omega \rightarrow \mathbb{R}^d$ the components $X_i(\omega)$ of which represent the number of units of the i -th asset in the portfolio, $i = 1, \dots, d$. If transaction costs are present, such risk measures are more appropriate than real-valued functions, which always represent a complete risk preference and thus cannot account for incomparable portfolios.

The theory of set-valued risk measures was initiated in [24] and systematically developed in [15] and [17]. We refer the reader to these references for further motivation and information. Here, we restrict ourself to the case of finite probability spaces and the question how the values of a set-valued risk measure can be computed. It will turn out that this leads to problems of type (P), hence one can apply the algorithms presented in Section 4.

The basic idea is as follows. The value of a set-valued risk measure at some random future portfolio X consists of initial deterministic portfolios u which can be given as deposits for the 'risky payoff' X , thus making the overall position 'risky payoff plus deposit' a non-risky one. It usually is not possible to use all assets as deposits, but rather a small subset including cash in a few currencies, bonds, gold or similar risk-free or low-risk assets. These 'eligible' assets are assumed to span the linear subspace M of \mathbb{R}^d with $1 \leq \dim M = m \leq d$. A typical example, already used in [24], is $\mathbb{R}^m \times \{0\}^{d-m}$, i.e., the first m assets are eligible.

Let (Ω, P) be a finite probability space and $N \geq 2$ be the number of elements in $\Omega = \{\omega_1, \dots, \omega_N\}$. We assume $p_n = P(\{\omega_n\}) > 0$ for all $n \in \{1, \dots, N\}$. The space of all multi-variate random variables $X: \Omega \rightarrow \mathbb{R}^d$ is denoted by $L_d^0(\Omega, P)$. A random variable $X \in L_d^0(\Omega, P)$ can be identified with an element $\hat{x} \in \mathbb{R}^{dN}$ through

$$\hat{x} = (X_1(\omega_1), \dots, X_d(\omega_1), X_1(\omega_2), \dots, X_d(\omega_N))^T \in \mathbb{R}^{dN}$$

and vice versa. Thus, the function $T: L_d^0(\Omega, P) \rightarrow \mathbb{R}^{dN}$ defined by $TX = \hat{x}$ is a linear bijection. If $A \subseteq L_d^0(\Omega, P)$ then we set $\hat{A} = \{\hat{x} \in \mathbb{R}^{dN} \mid \hat{x} = TX, X \in A\}$.

Let $K_0 \subseteq \mathbb{R}^d$ be a finitely generated convex cone satisfying $\mathbb{R}_+^d \subseteq K_0 \neq \mathbb{R}^d$. Such a 'solvency' cone models the market at initial time. We set $K_0^M = K_0 \cap M$ and $\mathcal{P}(M, K_0^M) = \{D \subseteq M \mid D = D + K_0^M\}$. A risk measure is a function $R: L_d^0(\Omega, P) \rightarrow \mathcal{P}(M, K_0^M)$ satisfying

$$\forall u \in M, \quad \forall X \in L_d^0(\Omega, P): R(X + u\mathbb{1}) = R(X) - u \quad (12)$$

where $\mathbb{1} : \Omega \rightarrow \mathbb{R}$ with $\mathbb{1}(\omega) = 1$ for all $\omega \in \Omega$ is the uni-variate random variable with constant value 1.

With R , we associate a risk measure $\hat{R} : \mathbb{R}^{dN} \rightarrow \mathcal{P}(M, K_0^M)$ by means of $\hat{R}(\hat{x}) = R(X)$ for $\hat{x} = TX$. Consequently, \hat{R} satisfies

$$\forall u \in M, \forall \hat{x} \in \mathbb{R}^{dN} : \hat{R}(\hat{x} + \hat{I}_d u) = \hat{R}(\hat{x}) - u, \quad (13)$$

where

$$\hat{I}_d = \begin{pmatrix} I_d \\ \vdots \\ I_d \end{pmatrix} \in \mathbb{R}^{dN \times d} \quad \text{and} \quad I_d = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{d \times d}.$$

The most common way to generate a risk measure is by means of a set $A \subseteq L_d^0(\Omega, P)$ of random variables which are considered to be ‘acceptable’ by the decision maker. The value of a risk measure generated by A then consists of all deterministic (available at time 0) portfolios $u \in M \subseteq \mathbb{R}^d$ which, when added to the uncertain future position X , make the overall position acceptable. Thus,

$$R_A(X) = \{u \in M \mid X + u\mathbb{1} \in A\}.$$

This functions indeed satisfies (12). Correspondingly,

$$R_{\hat{A}}(\hat{x}) = \{u \in M \mid \hat{x} + \hat{I}_d u \in \hat{A}\},$$

satisfies (13), and we have $R_{\hat{A}} = \widehat{R}_A$. Vice versa, with risk measures $R : L_d^0(\Omega, P) \rightarrow \mathcal{P}(M, K_0^M)$ and $\hat{R} : \mathbb{R}^{dN} \rightarrow \mathcal{P}(M, K_0^M)$ we associate the sets

$$A_R = \{X \in L_d^0(\Omega, P) \mid 0 \in R(X)\} \quad \text{and} \quad A_{\hat{R}} = \{\hat{x} \in \mathbb{R}^{dN} \mid 0 \in \hat{R}(\hat{x})\},$$

respectively. A basic fact about risk measures is a one-to-one correspondence between closed acceptance sets $A \subseteq L_d^0(\Omega, P)$ which satisfy $A + K_0^M \mathbb{1} \subseteq A$ and risk measures $R : L_d^0(\Omega, P) \rightarrow \mathcal{P}(M, K_0^M)$ with a closed graph by means of the above formulas. In particular, the relationships $\hat{A} = A_{R_{\hat{A}}}$ and $\hat{R} = R_{A_{\hat{R}}}$ hold true. See [17] for further details.

A risk measure R is called polyhedral if the associated risk measure \hat{R} is polyhedral, i.e., if

$$\text{gr } \hat{R} = \{(\hat{x}, u) \in \mathbb{R}^{dN} \times M \mid u \in \hat{R}(\hat{x})\},$$

is a polyhedral subset of $\mathbb{R}^{dN} \times \mathbb{R}^d$. The one-to-one correspondence between risk measures and their acceptance sets extends to the polyhedral case: $\hat{A} \subseteq \mathbb{R}^{dN}$ is polyhedral if, and only if, $R_{\hat{A}}$ is polyhedral, and \hat{R} is polyhedral if and only if $A_{\hat{R}}$ is polyhedral.

The above discussion leads to the following conclusion.

Remark 5.1 Since each polyhedral risk measure \hat{R} has the representation

$$\hat{R}(\hat{x}) = \{u \in M \mid \hat{x} + \hat{I}_d u \in A_{\hat{R}}\},$$

where $A_{\hat{R}}$ is a polyhedral set, the set $\hat{R}(\hat{x})$ is the upper image of a linear vector optimization problem. Indeed, if $A_{\hat{R}} \subseteq \mathbb{R}^{dN}$ has the H-representation $A_{\hat{R}} = \{\hat{y} \in \mathbb{R}^{dN} \mid \hat{B}\hat{y} \geq \hat{b}\}$ where

\hat{B} and \hat{b} are matrices of appropriate dimension then

$$\hat{R}(\hat{x}) = \left\{ u \in M \mid \hat{B}(\hat{x} + \hat{I}_d u) \geq \hat{b} \right\} = \left\{ u \in M \mid \hat{B} \hat{I}_d u \geq \hat{b} - \hat{B} \hat{x} \right\}.$$

Let $P \in \mathbb{R}^{d \times m}$ be a matrix with column vectors μ^1, \dots, μ^m forming a basis of M and define $B = \hat{B} \hat{I}_d P$, $b = \hat{b} - \hat{B} \hat{x}$. Then, observing that $\hat{R}(\hat{x}) + K_0^M = \hat{R}(\hat{x})$ and substituting $u = Pz$ we obtain that $\hat{R}(\hat{x})$ is the upper image of the problem

$$\text{minimize } P: \mathbb{R}^m \rightarrow M \text{ with respect to } \leq_{K_0^M} \quad \text{subject to } Bz \geq b.$$

However, this is just a theoretical result since in practice life is not as straightforward: Usually, the constraints describing \hat{A} involve a large number of auxiliary variables, and u is given as a linear function of those (see Example 5.3 below). Therefore, the algorithm presented in Section 4 is an appropriate tool to compute the values of a polyhedral set-valued risk measure because the dimension of the pre-image space usually is much greater than the dimension of the image space which is $\dim M = m \leq d$. Compare Examples 5.2 and 5.3 below.

In the following, we will discuss two examples which will be used for the numerical computations reported in Section 6.

Example 5.2 In worst case, the regulator/decision maker only accepts positions with non-negative components. Thus, the acceptance set is $A = (L_d^0)_+$ which is the set of all component-wise non-negative random variables. The market extension of the worst case risk measure, i.e., when trading is allowed, is related to the set of superhedging portfolios, see [27]. Its acceptance set in a one-period market is $A = (L_d^0)_+ + K_0 \mathbb{1} + L_d^0(K_T)$ where the cone K_0 and the random cone K_T model market conditions with a potential bid ask price spread at initial and terminal time, respectively, and $L_d^0(K_T) = \{X \in L_d^0 \mid \forall \omega \in \Omega: X(\omega) \in K_T(\omega)\}$. The cones $K_T(\omega)$ are also finitely generated convex cones satisfying $\mathbb{R}_+^d \subseteq K_T(\omega) \neq \mathbb{R}^d$ for all $\omega \in \Omega$.

The market extension of the worst case risk measure still is very conservative since a payoff X is acceptable only if there is a trading strategy such that its result, when added to X , is non-negative in all components in all possible scenarios, even those with a very small probability. Therefore, we introduce a ‘relaxed’ variant as follows.

Let $G \subseteq \mathbb{R}^d$ be a finitely generated convex cone with $\mathbb{R}_+^d \subseteq G \neq \mathbb{R}^d$ and consider the following acceptance set

$$A^{RWC} = \left\{ X \in L_d^0(\Omega, P) \mid \forall \omega \in \Omega: X(\omega) \in \left(-\epsilon + \mathbb{R}_+^d \right) \cap G \right\} + K_0 \mathbb{1} + L_d^0(K_T),$$

where $\epsilon \in \mathbb{R}^d$ such that $\epsilon_i \geq 0$ for all $i \in \{1, \dots, d\}$. Compared to the ‘true’ worst case risk measure, the set $(L_d^0)_+$ is replaced by ‘a little’ bigger set.

Thus, payoffs with ‘small’ negative components may still be considered acceptable, and the size of the risk related to such payoffs is controlled by ϵ and G . The cone G may serve as a conservative estimate of a market model which the regulator/supervisor thinks is robust enough to cover most market scenarios. For example, G can be chosen such that $P(G \subseteq K_T) \geq 1 - \alpha$ for some significance level $\alpha \in [0, 1]$. Then, the probability of a loss is bounded by α , and a potential loss (in physical units) is bounded by ϵ . Note that in the scalar case $d = m = 1$, the relaxed worst case risk measure reduces to the scalar worst case risk measure. The market extension of the relaxed worst case risk measure is given by

$$\begin{aligned} RWC(X) = \left\{ u \in M \mid z \in K_0, Z \in L_d^0(K_T), \forall \omega \in \Omega: \right. \\ \left. X(\omega) - z - Z(\omega) + u \in \left(-\epsilon + \mathbb{R}_+^d \right) \cap G \right\}, \end{aligned} \quad (14)$$

and can be seen as a relaxation of the superhedging set and thus as a good deal price bound of $-X$, see Example 6.6 for details. It is polyhedral (convex) as A^{RWC} is polyhedral (convex), but not sublinear. This is a new feature since the classical worst case risk measure is always sublinear.

In order to describe $RWC(\hat{x})$, let g^1, \dots, g^L be the generating vectors of the cone G and let \hat{G} be the $dN \times LN$ matrix which contains N blocks on its diagonal, where each block consists of the matrix with g^1, \dots, g^L as columns. Then

$$\forall n \in \{1, \dots, N\}: X(\omega_n) \in G \Leftrightarrow \exists \gamma \in \mathbb{R}_+^{LN}: \hat{x} = \hat{G}\gamma.$$

Similarly, let \hat{K}_T be the $dN \times J$ matrix which contains N blocks on its diagonal, where the first block consists of the matrix with the generating vectors of $K_T(\omega_1)$ as columns, the second block contains the generating vectors of $K_T(\omega_2)$ and so forth to the last block with the generating vectors of $K_T(\omega_N)$. J is the sum of the number of generating vectors of all K_T 's. Let \hat{K}_0 denote the matrix containing the I generating vectors of K_0 as columns. Then,

$$\begin{aligned} RWC(\hat{x}) &= \{Pz \mid \hat{x} + \hat{I}_d(Pz + \epsilon - \hat{K}_0 r) - \hat{K}_T s \in \mathbb{R}_+^{dN}, \\ &\quad \hat{I}_d(Pz - \hat{K}_0 r) - \hat{K}_T s - \hat{G}\gamma = -\hat{x}, z \in \mathbb{R}^m, \gamma \in \mathbb{R}_+^{LN}, r \in \mathbb{R}_+^I, s \in \mathbb{R}_+^J\}, \end{aligned}$$

with ordering cone K_0^M . Thus, the dimension of the pre-image space is $m + LN + I + J$ whereas the dimension of the image space is just m .

Example 5.3 The following set-valued function is a generalization of the scalar average value at risk (see [11, p. 210]) which is probably the most important and most studied example of a sublinear coherent measure of risk as introduced by [1]. Let $\alpha \in (0, 1]^d$. Define for $X \in L_d^0$

$$\begin{aligned} AV @ R_\alpha(X) &= \{\text{diag}(\alpha)^{-1} E[Z] - z \mid Z \in (L_d^0)_+, \\ &\quad X + Z - z \mathbb{1} \in K_0 \mathbb{1} + L_d^0(K_T), z \in \mathbb{R}^d\} \cap M \end{aligned} \quad (15)$$

where $\text{diag}(\alpha)^{-1}$ is the inverse of the diagonal matrix with the components of α on its main diagonal and zero elsewhere, and the cones $K_0, K_T(\omega)$ modeling the market conditions are as described above. Therefore this risk measure is also called the market extension of a simpler ‘regulator’ version, see [18]. We also refer to this paper for further motivation, interpretation and more details. It is immediately clear that $AV @ R_\alpha$ is not given in the form of R_A above, but it is a polyhedral convex (even sublinear) risk measure.

Its ‘hat’ variant can be derived as follows. Replace Z by $\hat{z} \in \mathbb{R}^{dN}$ and introduce auxiliary variables which admit to write $\hat{x} + \hat{z} - \hat{I}_d z$ as non-negative linear combination of the generating vectors of the cones K_0 and $K_T(\omega_n)$. Transform the objective into matrix form and get

$$AV @ R_\alpha(\hat{x}) = \{Px \mid Bx \geq b\} + K_0^M,$$

with appropriate matrices B, b and P where $K_0^M = K_0 \cap M$ as before. The dimension of the pre-image space is $d(N + 1) + I + J$ whereas the dimension of the image space is just m . It is worth mentioning that in the scalar case (i.e., without transaction costs) Rockafellar and Uryasev observed that the $AV @ R$ can be computed by solving a linear optimization problem, see [30].

6 Numerical examples

The algorithms have been implemented with MATLAB using the GNU Linear Programming Kit (GLPK) to solve the LPs and the CDDLIB package [5] for vertex enumeration. The graphics have been generated by JavaView³ and OpenOffice (Fig. 5). By a straightforward extension of the above results we can also solve linear vector optimization problems with constraints of the form

$$a \leq Bx \leq b \quad lb \leq x \leq ub, \quad (16)$$

where the components of a, b, lb, ub belong to $\mathbb{R} \cup \{-\infty, +\infty\}$. All examples were computed on a MacBook Pro with 2.26 GHz clock and 8 GB memory. We made use of the fact that all the LPs have a very similar form. This means that the matrix B does not need to be changed during the algorithm (except one line because η is not yet known at the beginning). This allows us to initialize LPs by appropriate basis solution of LPs solved in previous steps (warm starts).

In the following examples we provide tables with a few computational data, such as the total time and the number of LPs solved (# LPs). Note that we compute an ε -solution (\bar{S}, \bar{S}^h) of (P) and an ε -solution \bar{T} of (D*), compare Remark 4.10. We provide the cardinality $|\cdot|$ of the sets \bar{S} , \bar{S}^h and \bar{T} . Recall that we have $|\bar{S}^h| = 0$ whenever the problem is bounded. Note that $|\bar{S}|$ and $|\bar{T}|$ ‘correlate’ to the number of, respectively, vertices and facets of \mathcal{P} (but do not need to coincide exactly). One reason for possible differences is degeneracy as discussed in [26, Section 5.6], another one is numerical inaccuracy.

Further we denote by t_{\max} the maximum time used to solve one LP and by t_{aver} the average time to solve one LP. The quotient t_{\max}/t_{aver} indicates the impact of using warm starts. We start with two numerical examples from the literature.

Example 6.1 Shao and Ehrgott [32] used extended variants of Benson’s algorithm to solve linear vector optimization problems occurring in radio therapy treatment planning. We compute Example (PL) in [32] which has three objectives and a matrix B of size $1,211 \times 1,143$ with 153,930 nonzeros. The ordering cone is $C = \mathbb{R}_+^3$, and the problem is known to be bounded, which means that the first phase of our algorithms as well as the computation of η can be skipped. Further we set $c = (1, 1, 1)^T$.

The following table shows some results obtained by Algorithm 1. The second column in the table concerns the optional break command in the algorithm. One can observe that more LPs have to be solved when the break command is disabled. On the other hand, less vertex enumerations are required. This explains why the variant ‘no break’ is becoming faster than the ‘break’ variant when $\varepsilon > 0$ is small enough.

ε	Variant	Total time (s)	$ \bar{S} $	$ \bar{T} $	# LPs	t_{\max} (s)	t_{\max}/t_{aver}
0.3	Break	47	46	29	75	0.84	1.8
0.1	Break	91	104	61	165	0.87	2.0
0.05	Break	144	176	94	270	0.86	2.0
0.005	Break	1,596	1,456	597	2,053	0.84	1.9
0.3	No break	54	54	34	88	0.85	1.8
0.1	No break	114	134	78	212	0.84	1.9
0.05	No break	205	264	129	393	0.85	1.9
0.005	No break	1,411	1,945	804	2,749	0.84	1.9

³ By Konrad Polthier, <http://www.javaview.de>.

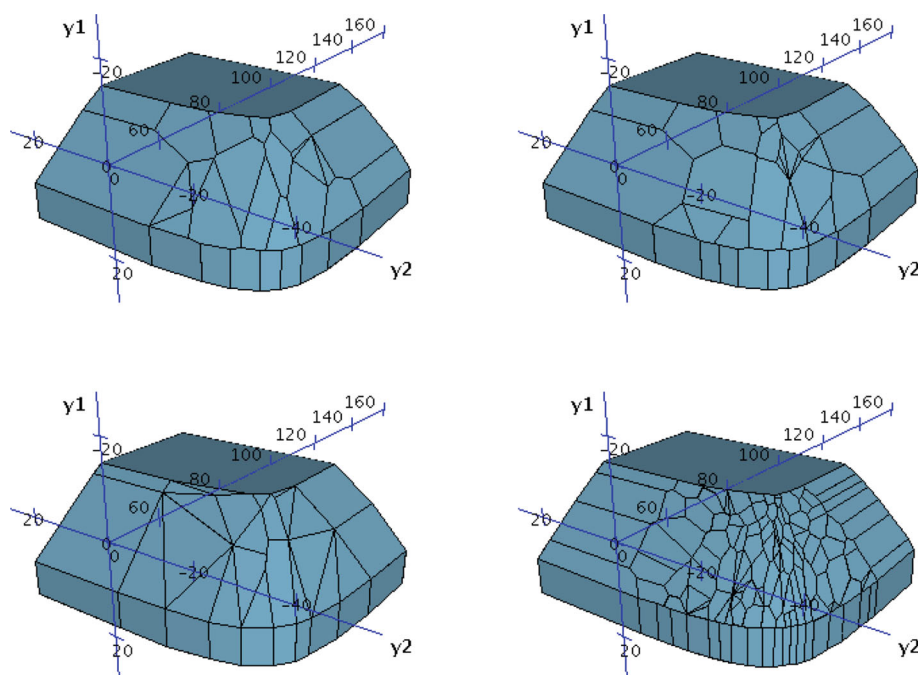


Fig. 1 Illustration of the upper image \mathcal{P} in Example 6.1; *top left*: primal algorithm, variant ‘break’, $\varepsilon = 0.3$; *top right*: primal algorithm, variant ‘no break’, $\varepsilon = 0.3$; *bottom left*: dual algorithm, variant ‘break’, $\varepsilon = 0.3$; *bottom right*: primal algorithm, variant ‘break’, $\varepsilon = 0.05$

Although we need less computational time than in [32], it is difficult to compare the results as we use a faster computer, a different (open source) LP solver, and we utilize warm starts. Moreover, in [32] an online vertex enumeration method is used, which is preferable if the number of vertices and facets of \mathcal{P} is large. Furthermore, our method yields the same approximation error as in [32] by less vertices and facets of \mathcal{P} . See Figure 1 for an illustration of part of the results.

Example 6.2 Ruszczyński and Vanderbei [31] developed a specialized parametric method for computing all minimizers of bi-criteria problems. Using intermediate results of the parametric simplex method, they solved in [31], for instance, a mean-risk model with a dense matrix B of size $6,161 \times 3,799$ having 4,435,919 nonzero entries. They pointed out that computing all the 5,017 minimizers takes only a little more time than solving one single LP. As this problem is known to be bounded, we can skip in our algorithms the first phase as well as the computation of η . For $c = (1, 1)^T$, our primal algorithm yields approximate solutions as shown in the following table.

ε	Total time (s)	$ \tilde{S} $	$ \tilde{T} $	# LPs	t_{\max} (s)	t_{\max}/t_{aver}
$\sqrt{2} \cdot 10^{-4}$	946	6	7	13	347	4.4
$\sqrt{2} \cdot 10^{-5}$	1,648	22	23	45	304	8.9
$\sqrt{2} \cdot 10^{-6}$	3,085	62	63	125	310	14.1

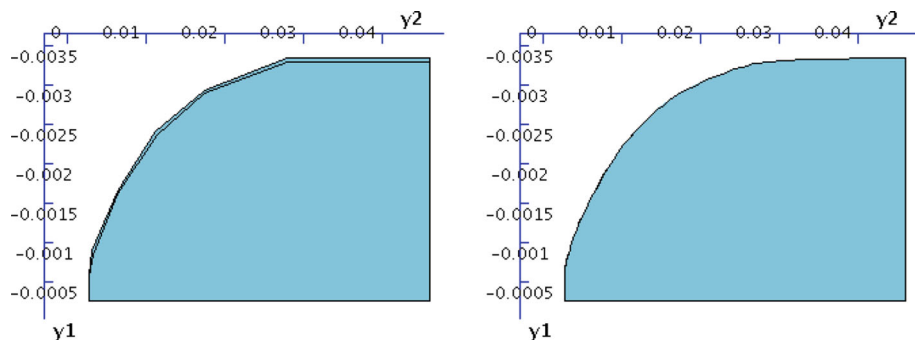


Fig. 2 Illustration of the upper image \mathcal{P} in Example 6.2 for $\varepsilon/\|c\| = 10^{-4}$ (inner and outer approximation, left) and for $\varepsilon/\|c\| = 10^{-6}$ (right)

We see that a ‘good’ approximation with $\varepsilon/\|c\| = 10^{-6}$ can be obtained in about ten times the time required to solve a single LP. This means our general method needs much more time for an ε -solution (compare Remark 4.10) than the parametric method for bounded bi-criteria problems in [31] needs for the exact solution. On the one hand, approximating solutions are often sufficient for a decision maker in practice, compare Fig. 2. On the other hand, we think that the ideas of the algorithm by Ruszczyński and Vanderbei are promising for further improvements of Benson type algorithms for arbitrary linear vector optimization problems.

The following three numerical examples refer to Example 5.3 in the previous section.

Example 6.3 Let us consider $d = 12$ assets, the first one is a risk-free USD bond with annual interest rate 5 %. Given is the vector of today’s asset prices, the vector of the expected returns and a covariance matrix for the other 11 correlated risky assets denoted in USD. Then, one can set up a one-period tree for the asset prices S_T with time horizon $T = 1$ year as in [25] to reflect the drift and covariance structure. The resulting number of scenarios is $N = 2^{d-1} = 2048$. We consider proportional transaction costs for the bond to be $\lambda_0 = 3\%$ and for the first risky asset (usually another currency) to be $\lambda_1 = 7\%$, for the second risky asset to be $\lambda_2 = 5\%$ and for all other risky assets to be 1 %. Then, the bid and ask prices of the assets are $(S_t^a)_i = (1 + \lambda_i)(S_t)_i$ and $(S_t^b)_i = (1 - \lambda_i)(S_t)_i$ for $i = 0, \dots, 11$ and $t \in \{0, T\}$. Furthermore, let us assume an exchange between any two risky assets can not be made directly, only via cash in USD by selling one asset and buying the other. Since the risk-free bond has strictly positive transaction costs λ_0 , the cones K_0 and $K_T(\omega_n)$ for $n = 1, \dots, N$ have $d(d-1) = 132$ generating vectors each. Thus $I = 132$ and $J = 270,336$.

We want to evaluate the risk of an outperformance option with physical delivery and maturity T . This option gives the right to buy the asset that performed best out of a basket of assets at a given strike price. Let the strike be $K = (1 + \lambda_1)(S_0)_1$. To normalize to today’s prices, let a vector g be defined as $(S_0^a)_1 = g_i(S_0^a)_i$ for $i \in \{1, \dots, 11\}$. The payoff X of the option is $-K$ in the risk free asset, g_i units of asset i for the smallest i satisfying $g_i(S_T^a)^i = \max_{j \in \{1, \dots, 11\}}(g_j(S_T^a)^j) \geq K$ and zero in the other assets. If $\max_{j \in \{1, \dots, 11\}}(g_j(S_T^a)^j) < K$ the payoff is the zero vector.

Let us calculate $AV @ R_\alpha(X)$ as described in Example 5.3 with significance levels

$$\alpha = (0.1, 0.08, 0.09, 0.1, 0.05, 0.05, 0.04, 0.05, 0.03, 0.04, 0.03, 0.04)^T.$$

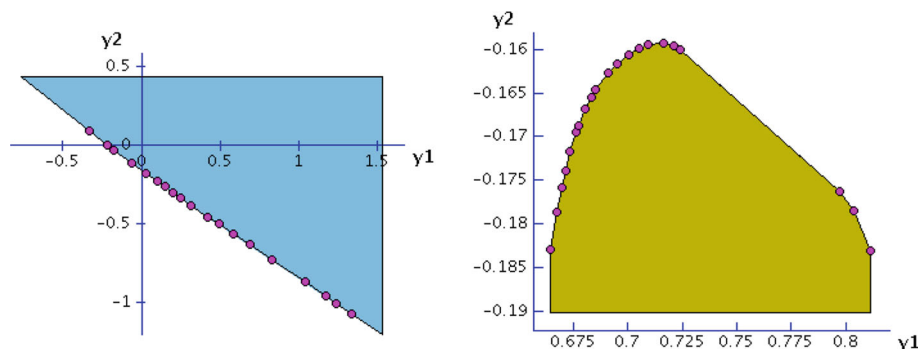


Fig. 3 Illustration of the upper image \mathcal{P} (left) and the lower image \mathcal{D}^* (right) in Example 6.3 for $\varepsilon = 10^{-4}$

As the space of eligible assets we choose the space spanned by the first and the second asset, i.e., $M = \mathbb{R}^2 \times \{0\}^{10}$. Formula (15) leads to a linear vector optimization problem with two objectives and constraints of the form (16) where the matrix B is of size $24,586 \times 295,056$. B is sparse having 1,150,986 nonzero entries. The ordering cone is K_0^M , which is strictly larger than \mathbb{R}_+^2 and is generated by two vectors. The vertices of $AV@R_\alpha(X)$ are minimal deposits in the bond and the second asset that compensate for the risk of X measured by $AV@R_\alpha$. The following table shows some computational data of the primal algorithm.

ε	Total time (s)	$ \bar{S} $	$ \bar{S}^h $	$ \bar{T} $	# LPs	t_{\max} (s)	t_{\max}/t_{aver}
10^{-4}	3,529	20	1	21	46	592	8.4
10^{-5}	4,716	47	1	48	100	671	17.1
10^{-6}	7,905	122	1	123	253	449	22.0

We can see that the problem is unbounded. In Fig. 3 the upper image \mathcal{P} and the lower image \mathcal{D}^* for $\varepsilon = 10^{-4}$ are shown. We observe that the vertices of \mathcal{P} are almost on a line and the lower image \mathcal{D}^* is more suitable to illustrate the example.

Example 6.4 Now consider $d = 11$ assets with a given correlation structure and all other input parameters as for the first 11 assets in Example 6.3 above. We have $N = 2^{d-1} = 1,024$ and the number of generating vectors of each cone K_0 and $K_T(\omega_n)$ for $n = 1, \dots, N$ is $d(d-1) = 110$. Consider a basket call option with physical delivery and strike price $K = \sum_{i=1}^{10} (S_0)_i$. If at maturity ($T = 1$ year) the value of the basket of risky assets is greater or equal to the strike, i.e., $\sum_{i=1}^{10} (S_T)_i \geq K$, then one would exercise the option and buy the 10 risky assets at strike K by delivering K times the bond, i.e., $X = (-K, 1, \dots, 1)^T$ in this case. If the value is less, the payoff vector is the zero vector. As the space of eligible assets we choose the space spanned by the first three assets, i.e., $M = \mathbb{R}^3 \times \{0\}^8$. The ordering cone is K_0^M , which is strictly larger than \mathbb{R}_+^3 and generated by six vectors. The linear vector optimization problem has three objectives and a matrix of size $11,272 \times 124,025$, which is sparse having 481,288 nonzero entries. For $\varepsilon = 10^{-3}$, the computational time of the primal algorithm was 1,748 s. The result is illustrated in Fig. 4. As the upper image \mathcal{P} is difficult to visualize (a polyhedron containing no lines but being ‘close’ to a halfspace) we only provide the lower image \mathcal{D}^* .

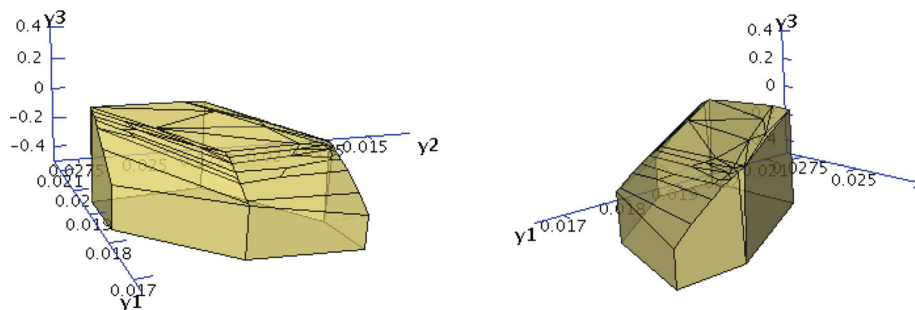
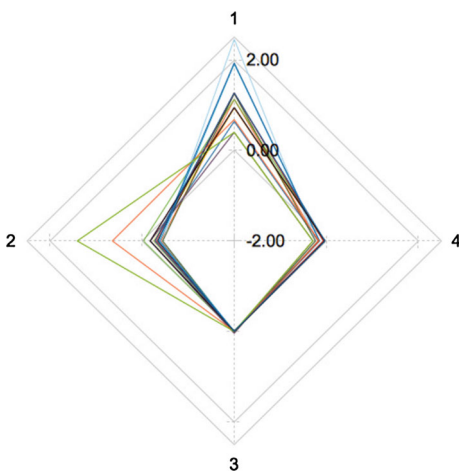


Fig. 4 The lower image \mathcal{D}^* in Example 6.4 for $\varepsilon = 10^{-3}$, two different view points

Fig. 5 Visualization of the 18 vertices of the upper image $\mathcal{P} \subseteq \mathbb{R}^4$ in Example 6.5 by a radar chart



Example 6.5 Consider $d = 10$ assets with a given correlation structure and all other input parameters as in Example 6.4 above. Let X be the payoff of an outperformance option with physical delivery as described in Example 6.3. As the space of eligible assets we choose the space spanned by the first four assets, i.e., $M = \mathbb{R}^4 \times \{0\}^6$. The corresponding linear vector optimization problem has four objectives and a matrix of size $5,126 \times 51,300$ with 197,638 nonzero entries. The ordering cone is K_0^M , which is strictly larger than \mathbb{R}_+^4 and generated by 12 vectors. Then, $AV @ R_\alpha(X)$ with α as in Example 6.4, obtained as the upper image of linear vector optimization problem computed with the primal algorithm and $\varepsilon = 10^{-2}$, has 18 vertices and 12 extreme directions. The vertices of \mathcal{P} are visualized by a radar chart in Fig. 5.

The following numerical example refers to Example 5.2 in the previous section.

Example 6.6 Let us consider $d = 9$ assets with a given correlation structure, all other input parameters as in Example 6.4 above (i.e., $m = 3$), and the same basket option (basket call) with payoff X as in Example 6.4. We want to calculate $RWC(-X)$, the relaxed worst case risk measure at $-X$, as described in Example 5.2 with parameter $\epsilon_i = 500$ for $i \in \{3, \dots, 9\}$ and zero otherwise. The cone G can be seen as a worst case solvency cone and is chosen to be a conservative modification of K_0 , where λ is replaced by the larger transaction costs $\lambda_{wc} = \lambda + 0.2$.

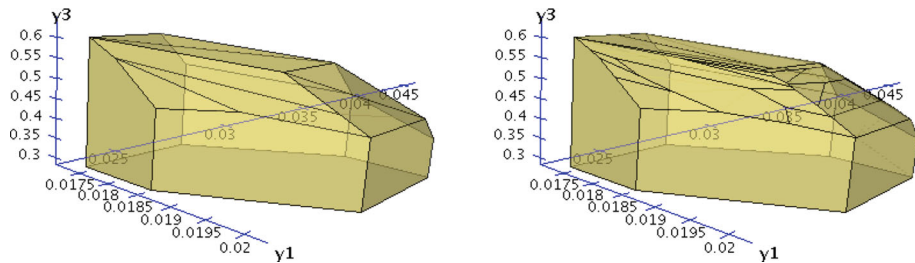


Fig. 6 The lower image \mathcal{D}^* in Example 6.6 for $\varepsilon = 10^{-2}$ (left) and $\varepsilon = 10^{-3}$ (right) computed with the dual algorithm

$RWC(-X)$ corresponds to an upper good deal bound as it is a relaxed version of the set of superhedging portfolios. By considering certain small risks controlled by ϵ and G as acceptable, the scalar superhedging price of 34.942464 units of bond is reduced to 34.830995 units of bond for $RWC(-X)$. The linear vector optimization problem to calculate $RWC(-X)$ has 3 objectives and a matrix B of size $4,608 \times 36,939$, which is sparse with 185,856 nonzero entries. The above prices in units of bond were obtained by solving linear vector optimization problems with both the primal and dual algorithm for $\varepsilon = 10^{-4}$.

The following table shows some computational data and a comparison of the primal and dual algorithm. In Fig. 6, parts of the results are visualized.

ε	Variant	Total time (s)	$ \bar{S} $	$ \bar{T} $	# LPs	t_{\max} (s)	t_{\max}/t_{aver}
10^{-2}	Primal	113	13	12	37	8.8	3.8
10^{-3}	Primal	239	68	37	123	9.0	6.8
10^{-4}	Primal	506	153	82	308	8.8	8.6
10^{-2}	Dual	86	7	20	37	5.8	3.8
10^{-3}	Dual	193	30	73	113	8.2	4.8
10^{-4}	Dual	404	74	136	256	5.8	5.1

Acknowledgements We thank Dr Lizhen Shao for providing the data of Example 6.1 taken from [32] and we thank Professor Robert Vanderbei for supplying the data of Example 6.2 taken from [31].

References

- Artzner, P., Delbaen, F., Eber, J.-M., Heath, D.: Coherent measures of risk. *Math. Finance* **9**(3), 203–228 (1999)
- Barber, C., Dobkin, D.P., Huhdanpaa, H.: The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **22**(4), 469–483 (1996)
- Benson, H.: Further analysis of an outcome set-based algorithm for multiple-objective linear programming. *J. Optimiz. Theory Appl.* **97**(1), 1–10 (1998a)
- Benson, H.: An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *J. Global Optimiz.* **13**, 1–24 (1998b)
- Bremner, D., Fukuda, K., Marzetta, A.: Primal—dual methods for vertex and facet enumeration. *Discrete Comput. Geom.* **20**(3), 333–357 (1998)
- Csirmaz, L.: Using multiobjective optimization to map the entropy region of four random variables. <http://eprints.renyi.hu/66/> (2013)

7. Ehrgott, M.: Solving multiobjective linear programmes—from primal methods in decision space to dual methods in outcome space. In: Book of Abstracts, Second South Pacific Conference on Mathematics, Noumea, New Caledonia, August 30th to September 3rd, (2010). <http://pages.univ-nc.nc/bonnel/spcm-2010/confspcm10.htm>
8. Ehrgott, M., Löhne, A., Shao, L.: A dual variant of Benson's outer approximation algorithm. Report 654, University of Auckland School of Engineering (2007)
9. Ehrgott, M., Shao, L., Schöbel, A.: An approximation algorithm for convex multi-objective programming problems. *J. Global Optimiz.* **50**(3), 397–416 (2011)
10. Ehrgott, M., Löhne, A., Shao, L.: A dual variant of Benson's outer approximation algorithm. *J. Global Optimiz.* **52**(4), 757–778 (2012)
11. Föllmer, H., Schied, A.: Stochastic Finance, Extended Edition. Walter de Gruyter, Berlin (2011)
12. Hamel, A.H., Löhne, A.: Lagrange duality in set optimization (submitted) (2012). arXiv:1207.4433.
13. Hamel, A.H.: A duality theory for set-valued functions I: Fenchel conjugation theory. *J Set-Valued Var. Anal.* **17**, 153–182 (2009)
14. Hamel, A.H.: A Fenchel–Rockafellar duality theorem for set-valued optimization. *Optimization* **60**(7–9), 1023–1043 (2011)
15. Hamel, A.H., Heyde, F.: Duality for set-valued measures of risk. *SIAM J. Financ. Math.* **1**, 66–95 (2010)
16. Hamel, A.H., Heyde, F., Löhne, A., Tammer, C., Winkler, K.: Closing the duality gap in linear vector optimization. *J. Convex Anal.* **11**(1), 163–178 (2004)
17. Hamel, A.H., Heyde, F., Rudloff, B.: Set-valued risk measures for conical market models. *Math. Financ. Econ.* **5**, 1–28 (2011)
18. Hamel, A.H., Rudloff, B., Yankova, M.: Set-valued average value at risk and its computation. *Math. Financ. Econ.* **7**(2), 229–246 (2013)
19. Heyde, F.: Geometric duality for convex vector optimization problems. (submitted) (2011). arXiv:1109.3592v1.
20. Heyde, F., Löhne, A.: Geometric duality in multiple objective linear programming. *SIAM J. Optimiz.* **19**(2), 836–845 (2008)
21. Heyde, F., Löhne, A.: Solution concepts in vector optimization: A fresh look at an old story. *Optimization* **60**(12), 1421–1440 (2011)
22. Heyde, F., Löhne, A., Tammer, C.: The attainment of the solution of the dual program in vertices for vectorial linear programs. In: Barichard, V. et al. (Eds.) Multiobjective programming and goal programming. Theoretical results and practical applications. Lecture Notes in Economics and Mathematical Systems vol. 618. Springer, pp. 13–24 (2009a)
23. Heyde, F., Löhne, A., Tammer, C.: Set-valued duality theory for multiple objective linear programs and application to mathematical finance. *Math. Methods Oper. Res.* **69**(1), 159–179 (2009b)
24. Jouini, E., Meddeb, M., Touzi, N.: Vector-valued coherent risk measures. *Finance Stoch.* **8**, 531–552 (2004)
25. Korn, R., Müller, S.: The decoupling approach to binomial pricing of multi-asset options. *J. Comput. Finance* **12**(3), 1–30 (2009)
26. Löhne, A.: Vector Optimization with Infimum and Supremum. Springer, Berlin (2011)
27. Löhne, A., Rudloff, B.: An algorithm for calculating the set of superhedging portfolios and strategies in markets with transaction costs. (submitted) (2011). arXiv:1107.5720v1.
28. Löhne, A., Schrage, C.: An algorithm to solve polyhedral convex set optimization problems. *Optimization* **62**(1), 131–141 (2013)
29. Luc, D.T.: On duality in multiple objective linear programming. *Eur. J. Oper. Res.* **210**, 158–168 (2011)
30. Rockafellar, R.T., Uryasev, S.P.: Optimization of conditional value-at-risk. *J. Risk* **2**, 21–42 (2000)
31. Ruszczyński, A., Vanderbei, R.J.: Frontiers of stochastically nondominated portfolios. *Econometrica* **71**(4), 1287–1297 (2003)
32. Shao, L., Ehrgott, M.: Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. *Math. Methods Oper. Res.* **68**(2), 257–276 (2008a)
33. Shao, L., Ehrgott, M.: Approximating the nondominated set of an MOLP by approximately solving its dual problem. *Math. Methods Operat. Res.* **68**(3), 469–492 (2008b)